REMARKS

The Office action has been carefully considered. The Office action rejected claims 1, 3-9, 11-28, and 31-37 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,430,542 to Moran ("Moran"). Applicants respectfully disagree.

Applicants thank the Examiner for the interview held (by telephone) on February 15, 2006. During the interview, the Examiner and applicants' attorney discussed the claims with respect to the prior art. The essence of applicants' position is incorporated in the remarks below.

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is generally directed toward a financial or other planning system and method in which hierarchically arranged objects are created and maintained to form a plan. The hierarchical arrangement enables objects to be dependent on other objects, while within the objects are fields that can be related to other fields, *e.g.*, dates, dollar amounts, interest rates and so on. Significantly, once the user establishes one or more hierarchical dependencies, the user of the system and method need not be concerned with the dependencies and/or relationships among objects and fields, but rather may simply select elements and enter data for those elements, and thereafter let the objects of the system and method handle the dependencies. Thus, unlike simple programming techniques, a user may simply respond to questions via a user interface, fills in information and/or makes selections related to the plan. The system then writes the proper

information into the hierarchically arranged objects for the user, and manages the relationships for the user. A planning engine runs a simulation based on the data in the objects.

For example, a user may choose to enter a home mortgage balance due that represents a total dollar amount owed on a home. Then, the user may enter a second piece of information such as an amount for a monthly savings deposit that represents an amount of money that the user intends to save each month in a bank account. With these two inputted data stores in place, the user may then choose to enter an additional input (i.e., a third data entry) that defines a hierarchical relationship between the first and second data stores such that a value in the second data store is at least partially based on the first data store as a result of the hierarchical relationship. One example of a hierarchical relationship may be only depositing the savings amount in the bank account each month once the mortgage balance drops below a specified balance. Another example may be depositing a first amount of savings in a bank account each month at a first level until the savings balance reaches a threshold balance and then applying a maximum amount of funds toward the mortgage balance. In any case, once established, the user need not be concerned with the hierarchical relationships if the user chooses to change values associated with the underlying objects.

The combination of user-definable and user-reconfigurable hierarchical objects relationships and relative field values allows a great deal of flexibility in creating what may be a very complex data system, which can then be used to calculate the results of the user's financial plan over time, as well as making it fairly

straightforward for the user to make changes to and update a plan. Moreover, via simple interaction with the user interface, a user can selectively disable objects and/or fields, which automatically disables additional objects and fields that are dependent on the directly disabled ones. This facilitates the running of various "what-if" type simulations, to determine the expected consequences of various possible actions.

Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

Turning to the claims, independent claim 1, as amended, recites a computer-readable medium having computer-executable instructions, comprising, receiving input of a value corresponding to a first field of a first object that maintains plan data, receiving additional input corresponding to a second field of a second object that maintains plan data, receiving input that defines a hierarchical relationship between the first and second objects such that a value in the second field is at least partially based on the first field as a result of the hierarchical relationship, such that the hierarchical relationship is definable by the user and reconfigurable by the user with regard to the relationship between the first and second objects, developing a plan by running a simulation on objects that maintain the plan data including the first and second objects, receiving input of a new value for the first field, and developing a new plan by running a simulation on objects that maintain the plan data, including the first and second objects, in which in the new plan, the new value changes the information in the second field.

The Office action rejected claim 1 as anticipated by Moran. More specifically, the Office action contends that Moran teaches receiving input of a value corresponding to a first field of a first object that maintains plan data. Column 16, lines 25-35 of Moran is referenced. Further, the Office action contends that Moran teaches receiving additional input corresponding to a second field of a second object that maintains plan data. Column 20, lines 20-40 of Moran is referenced. Still further, the Office action contends that Moran teaches receiving input that defines a hierarchical relationship between the first and second objects such that a value in the second field is at least partially based on the first field as a result of the hierarchical relationship. Column 10, lines 47-60 of Moran is referenced. Further yet, the Office action contends that Moran teaches developing a plan by running a simulation on objects that maintain the plan data including the first and second objects. Column 21, line 62 to column 23, line 55 of Moran is referenced. Finally, the Office action contends that Moran teaches receiving input of a new value for the first field, and developing a new plan by running a simulation on objects that maintain the plan data, including the first and second objects, in which in the new plan, the new value changes the information in the second field. Again, column 16, lines 25-35 of Moran is referenced. Applicants respectfully disagree.

Moran teaches, generally, a financial planning system that allows an advisor to take advantage of certain dependencies between input data, such as, for example, an inheritance triggered by the death of someone or an annual contribution to a savings fund on a person's birthday. As specifically cited by the

Office action, column 20, lines 20-52 details a method by which a first data entry consists of one or more monthly expenses incurred by a person. A second data entry corresponds to a date of death of this person. As such, after death, this person is no longer going to incur monthly expenses and the monthly expenses drop to zero upon the date of death based on the predetermined relationship between the monthly expenses data store and the date of death data store.

However, this relationship between the first and second data entries is not described anywhere in Moran as reconfigurable or user-accessible. That is, the financial software, as a whole, allows for specific definitions of relationships between two data fields, but does not allow that the relationship itself be user-defined or user-configurable. Instead, Moran goes into great detail to describe the ability of the program to allow for an adjustment to all other data stores, such as monthly expenses, the income of other's around the deceased, and other investment goal assumptions. It is clear though, that the original relationship between the first and second entries or any other relationship in the method of Moran described above is not inputted by the user. Furthermore, Moran does not teach in any capacity being able to input a relationship between two data stores.

In contrast, claim 1 recites receiving input that defines a hierarchical relationship between the first and second objects. As such, in the method of claim 1, a user may define the hierarchical relationship between the first and second objects by inputting data. In this manner, the relationship itself is user-definable and/or reconfigurable as opposed to merely manipulating the data entry fields surrounding the relationship as is the case in Moran. Simply put, Moran does not

teach receiving input that defines a hierarchical relationship between the first and second objects.

This difference may be better illustrated by using a second example from the text of Moran. Moran goes into specific detail at columns 15-16 about presenting goal icons that may be selected by a user. For example, a user may select an icon representing the goal of "Joshua's education" and this will bring up a data entry screen. A user may enter data into fields representing such concepts as what college to attend and how much money will be required for attendance. The example goes on further to allow for a change in goals. Thus, a hypothetical client (Craig Burke in their example) may set up a contingency goal in the event that the client unexpectedly passes away. Thus, a first goal of having Joshua attending Harvard remains the current goal so long as Craig Burke does not die. If, however, Craig Burke dies, then Joshua will attend Kansas State University because the contingency goal was based on a conditional event (*i.e.,* Craig Burke dying). This contingency goal, however, is a system-defined relationship as Kansas State University will become the college of choice if and only if Craig Burke dies. This conditional event is not user-definable and is not reconfigurable. The name of the university may be changed, but the relationship between Craig Burke dying and Joshua attending a different university remains outside of the control of any user or advisor as it is a pre-defined relationship that is part of the financial software.

In contrast, using the same example, the present invention provides far more flexibility for a user to define the contingency plan. Using the present invention, a user may enter a first piece of data, such as establishing an account

for Joshua's education. The user may then further define a second data field as a list of universities that may be hierarchically arranged according to cost. A first level entry may be Local Vo-Tech University, a mid level entry may be Kansas State University, and a high level entry may be Harvard University. With this data in place, the user may enter a third piece of data that establishes a hierarchical relationship between Joshua's education account (the first data entry) and Joshua's school (the second data entry) such that as the balance in Joshua's education account surpasses different levels, the school choice trends toward more expensive schools.

It is this third data entry that the system of Moran cannot accomplish. The system of Moran may change the first and second data fields (*i.e.*, Joshua's education account balance and Joshua's school of choice) but it cannot change the relationship between the two once established (*i.e.*, if Craig Burke dies, then Joshua will go to the contingency school – whatever school happens to be in that data field). In the present invention, if the user so chooses, the user may define the relationship in any manner including, for example, if Joshua's education account reaches a very large balance, not choosing a school at all. The fact remains that the third data entry is user-definable and user-reconfigurable.

The recitations of claim 1 support this interpretation. Specifically, a key difference between Moran and the recitations of claim 1 is that the method recited in claim 1 provides for receiving three different inputs regarding a single relationship within the plan data. Claim 1 recites receiving input of a value corresponding to a first field of a first object, receiving additional input

corresponding to a second field of a second object, and receiving input that defines a hierarchical relationship between the first and second objects. The method taught by Moran does not encompass receiving these inputs. Contrarily, Moran merely teaches a method of receiving inputs corresponding to data for a first data store and a second data store. However, all subsequent data entries are directed toward either entering data for other data stores (*i.e.* not the first or the second described here) or are directed toward changes the data in the first and second data stores. Nowhere in the teachings of Moran is there disclosed a method or mechanism for inputting a hierarchical relationship between the first and second objects.

For at least the foregoing reasons, applicants submit that claim 1 is allowable over the prior art of record.

Applicants respectfully submit that dependent claims 3-9, 11-16, and 34 by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 1 and consequently includes the recitations of independent claim 1. As discussed above, Moran, fails to disclose the recitations of claim 1 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 1 noted above, each of these dependent claims includes additional patentable elements.

For example, claim 16 recites receiving input information that includes synchronizing plan elements with data from another program. The Office action contends that this recitation is taught by Moran's disclosure of receiving input from a user interface. Column 10, lines 60-65 of Moran is referenced. Applicants

respectfully disagree. A user interface, such as a keyboard or a mouse, is clearly not the same as another program. Even if one were to construe a user interface as being controlled by a driver program, the driver program certainly is not cognizant of financial plan elements and thus cannot possibly be used in synchronizing plan elements with data from another program as recited in claim 16. For at least this additional reason, applicants submit that claim 16 is allowable over the prior art of record.

Turning to the next independent claim, amended claim 17 recites in a computer system, a method of organizing information related to a plan, comprising, providing access to a limited number of objects to a user, each object having fields therein for maintaining plan information, receiving first user input information including a value associated with a first field of a first object, receiving second user input information associated with a second field of a second object, the second input information specifying a relationship of the second field with the first field such that the relationship is definable by the user and reconfigurable by the user with regard to the relationship between the first and second objects, disabling at least one object, and developing a plan including running a simulation that excludes each disabled object.

The Office action rejected claim 17 as anticipated by Moran. More specifically, the Office action contends that Moran teaches each recitation in claim 17 and cites the same rationale as was previously cited with respect to the rejection of claim 1. Applicants respectfully disagree.

As discussed above, Moran teaches, generally, a financial planning system that allows an advisor to take advantage of certain dependencies between input data. In particular, column 20, lines 20-52 detail a method by which a first data entry is one or more monthly expenses incurred by a person. A second data entry corresponds to a date of death of this person. As such, because this person is no longer going to incur monthly expenses, the monthly expenses drop to zero upon the date of death. However, this relationship between the first and second data entries is not described in Moran as user-accessible, user-definable or reconfigurable. It is clear though, that the original relationship between the first and second entries in the method of Moran described above is not inputted by the user nor is this relationship able to be changed in any manner.

In contrast, claim 17 recites receiving second user input information associated with a second field of a second object, the second input information specifying a relationship of the second field with the first field such that the relationship is definable by the user and reconfigurable by the user with regard to the relationship between the first and second objects. As such, in the method of claim 17, a user may define and even reconfigure a relationship between the first and second objects by inputting data. In this manner, the relationship itself is adjustable as opposed to the data entry fields surrounding the relationship as is the case in Moran. Simply put, Moran does not teach receiving input that defines a relationship between the first and second objects that is user-definable and user-reconfigurable. Applicants submit that claim 17 is allowable over the prior art of record for at least the foregoing reasons.

Applicants respectfully submit that dependent claims 18-19 and 35, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 17 and consequently includes the recitations of independent claim 17. As discussed above, Moran fails to disclose the recitations of claim 17 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 17 noted above, each of these dependent claims includes additional patentable elements.

Turning to the next independent claim, amended claim 20 recites a system for outputting a plan, comprising, a user interface for presenting a limited number of plan objects to a user and for receiving data for a first field of a first plan object and data for a second field of a second plan object, the data of the second field having a value linked to the data of the first field via a hierarchical relationship between the first and second objects, such that the hierarchical relationship is definable by a user and reconfigurable by the user with regard to the relationship between the first and second objects, the user interface further providing a mechanism that allows plan objects to be selectively disabled, and a planner engine for developing a plan by running a simulation on plan objects while excluding any disabled plan objects.

The Office action rejected claim 20 as anticipated by Moran. More specifically, the Office action contends that Moran teaches each recitation in claim 20 and cites the same rationale as was previously cited with respect to the rejection of claim 1. Applicants respectfully disagree.

Once again, Moran teaches, generally, a financial planning system that allows an advisor to take advantage of certain dependencies between input data.

In particular, column 20, lines 20-52 detail a method by which a first data entry is one or more monthly expenses incurred by a person. A second data entry corresponds to a date of death of this person. As such, because this person is no longer going to incur monthly expenses, the monthly expenses drop to zero upon the date of death. However, this relationship between the first and second data entries is not described in Moran as adjustable or user-accessible. It is clear though, that the original relationship between the first and second entries in the method of Moran described above is not inputted by the user nor is this relationship able to be changed in any manner.

In contrast, claim 20 recites a user interface for receiving data for a second field of a second plan object, the data of the second field having a value linked to the data of the first field via a hierarchical relationship between the first and second objects, such that the hierarchical relationship is definable by a user and reconfigurable by the user with regard to the relationship between the first and second objects. As such, in the system of claim 20, a user, through the user interface, may define and even reconfigure a relationship between the first and second objects by inputting data. In this manner, the relationship itself is adjustable as opposed to the data entry fields surrounding the relationship as is the case in Moran. Simply put, Moran does not teach a user interface capable of receiving input that defines a relationship between the first and second objects. Applicants submit that claim 20 is allowable over the prior art of record for at least the foregoing reasons.

Applicants respectfully submit that dependent claims 21-28, 31-33, and 37 by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 20 and consequently includes the recitations of independent claim 20. As discussed above, Moran fails to disclose the recitations of claim 20 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 20 noted above, each of these dependent claims includes additional patentable elements.

For example, claim 24 recites that the second field represents a date conditional on the amount represented in the first field. Nowhere in Moran is there any teaching of a field that expresses a date that is conditional on an amount. Moran discloses inputting dates such as date of death to then manipulate other fields based on the date. It is simply counter-intuitive for Moran to teach the reverse, *e.g.*, that a person may now die as soon as a savings account reaches $100,000. For at least this additional reason, applicants submit that claim 24 is allowable over the prior art of record.

Turning to the last independent claim, amended claim 36 recites a computer-readable medium having computer-executable instructions, comprising providing access to a limited number of objects to a user, each object having fields therein for maintaining plan information, receiving first user input information including a value associated with a first field of a first object, receiving second user input information associated with a second field of a second object, the second input information specifying a relationship of the second field with the first field, such that the relationship is definable by the user and reconfigurable by the user

with regard to the relationship between the first and second objects, disabling at least one object, and developing a plan including running a simulation that excludes each disabled object.

The Office action rejected claim 36 as anticipated by Moran. More specifically, the Office action contends that Moran teaches each recitation in claim 36 and cites the same rationale as was previously cited with respect to the rejection of claim 1. Applicants respectfully disagree.

As has been put forth above, Moran teaches, generally, a financial planning system that allows an advisor to take advantage of certain relationship between input data but does not allow the relationship itself between the first and second data entries to be adjustable or user-accessible. It is clear, that the system-defined relationship between the first and second entries in the system of Moran is not inputted by the user nor is this relationship able to be changed in any manner.

In contrast, claim 36 recites receiving second user input information associated with a second field of a second object, the second input information specifying a relationship of the second field with the first field, such that the relationship is definable by the user and reconfigurable by the user with regard to the relationship between the first and second objects. As such, in the computer-readable medium of claim 36, a user may define a hierarchical relationship between the first and second objects by inputting data via the computer-readable medium. In this manner, the relationship itself is user-definable and user-reconfigurable. This is different than merely being able to re-enter data into data fields associated with a system-defined relationship as is the case in Moran.

Simply put, Moran does not teach receiving input from a user that defines a relationship between the first and second objects. Applicants submit that claim 36 is allowable over the prior art of record for at least the foregoing reasons.

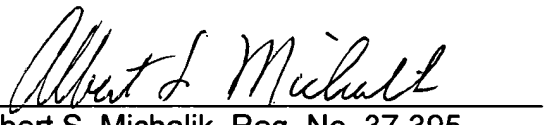
For at least these additional reasons, applicants submit that all the claims are patentable over the prior art of record. Reconsideration and withdrawal of the rejections in the Office action is respectfully requested and early allowance of this application is earnestly solicited.

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1, 3-9, 11-28, and 31-37 are patentable over the prior art of record, and that the application is in good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,



Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)